It Starts with iGaze: Visual Attention Driven Networking with Smart Glasses

Lan Zhang[1], Xiang-Yang Li[1,2], Wenchao Huang[3], Kebin Liu[1], Shuwei Zong[4]
Xuesi Jian[2], Puchun Feng[1], Taeho Jung[2], Yunhao Liu[1]
[1]School of Software and TNLIST, Tsinghua University, China
[2]Department of Computer Science, Illinois Institute of Technology, USA
[3]Department of Computer Science, University of Science and Technology of China
[4]Department of Computer Science, Suzhou Institute for Advanced Study, China
lan@greenorbs.com, xli@cs.iit.edu, yunhao@greenorbs.com

ABSTRACT

In this work, we explore a new networking mechanism with smart glasses, through which users can express their interest and connect to a target simply by a gaze. Doing this, we attempt to let wearable devices understand human attention and intention, and pair devices carried by users according to such attention and intention. To achieve this ambitious goal, we propose a proof-of-concept system **iGaze**, a visual attention driven networking suite: an **iGaze** glass (hardware), and a networking protocol **VAN** (software). Our glass, **iGaze** glass, is a low-cost head-mounted glass with a camera, orientation sensors, microphone and speakers, which are embedded with our software for visual attention capture and networking. A visual attention driven networking protocol (**VAN**) is carefully designed and implemented. In **VAN**, we design an energy efficient and highly accurate visual attention determination scheme using single camera to capture user's communication interest and a double-matching scheme based on visual direction detection and Doppler effect of acoustic signal to lock the target devices. Using our system, we conduct a series of trials for various application scenarios to demonstrate the effectiveness of our system.

Categories and Subject Descriptors

C.3 [**Special-purpose and application-based systems**]: Real-time and embedded systems

General Terms

Design, Experimentation, Performance

Keywords

Smart Glasses; Attention Driven Networking; Device Paring; Gaze Tracking

1. INTRODUCTION

Emerging wearable computing devices attract extensive attention worldwide. Smart glasses, *e.g.*, Google Glass and Vuzix Smart Glasses, are computerized eyeglasses. They not only are miniature video monitors, but also possess enhanced data processing functionalities. Besides, modern smart glasses are equipped with various sensors, such as gyroscope, accelerometer and GPS, to enrich their capabilities.

Many efforts have been devoted to improve human interfaces on or using smart glasses, *e.g.* controlling smart glasses using relatively simple voice commands, gesture control with a tilt of your head, photo and video capture, viewing and posting social status updates, and navigation. Some smart glasses also include features like augmented reality overlays. However smart glasses are still in the early stage and looking for revolutionary applications.

In this work, we explore a new networking mode with smart glasses that may enable a whole new set of applications. That is, smart glasses are supposed to provide users a novel way to communicate with surrounding physical and cyber world which is driven by their visual attention. We have a vision of the future world which has already been imagined in science fiction films, where people wearing smart glasses can establish network connections with their interest targets simply by a gaze. People can directly send messages to others without exchanging contact information explicitly, retrieve rich information about an object in the field of view[1] freely using their eye gazes. For example, in an academic conference, researchers can exchange information such as research interests by simply looking at each other. In a museum, visitors can obtain the detailed descriptions of an artwork by staring at it. Also, many new features can be brought to the game design, *e.g.*, real-scene role-playing games and interactive games with users in your sight. Moreover, the advertisements can also be significantly enhanced with such augmented reality. When people look at a signboard or product, relevant promotion information can be sent to the potential customers' smart glasses immediately. As a result, the store can provide more precise targeted ads and take advantage of the big impulse buying market [27].

These requirements can hardly be fulfilled by existing network solutions, such as using pre-exchanged identifiers, interest-based methods, gesture-based methods and image recognition. Most traditional networking systems rely on pre-exchanged network identifiers, *e.g.*, email, IP and web account, to start communications. Some interest based systems provide customized information services using the user's profile and mining results of his/her historical behaviors, *e.g.*, [13]. A few gesture-based device pairing approaches [22, 28] attempt to pair devices in proximity. However, those efforts are targeting handhold smart devices, *e.g.*, smart phones. Their requirements for phone displacement (> 20cm) [22]

[1]Using only proximity to retrieve information of an object may overwhelm a user with objects that are not in her vision focus.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MobiCom'14, September 7-11, 2014, Maui, Hawaii, USA.
Copyright 2014 ACM 978-1-4503-2783-1/14/09 ...$15.00.
http://dx.doi.org/10.1145/2639108.2639119.

and movement speed ($2 \sim 6m/s$) [28] are hard to achieve using head-mounted glasses. Besides, certain augmented reality could also be enabled with a world camera in a visual fashion, *e.g.*, using QR codes or object recognition techniques. But, to utilize the QR code, there are restrictions for scanning distances and angles to the target. For example, with a high quality camera (as the one of iPhone 5s) and a big QR code ($10cm \times 10cm$), the code is only scannable within a distance less than 3m and a view angle less than 30°. Worse cameras make the restrictions much stricter. Object recognition usually requires extensive training and is vulnerable to environment noise. It cannot distinguish two objects with the same appearance. More importantly, there could be multiple objects in the user's field of view. Without understanding the true interested target of the user, it is hard to determine which object to connect.

To achieve this future vision, we propose a new way to establish network connections using smart glasses, by which the user can connect to a target simply by a gaze. To this end, we address the following challenges.

1) How to accurately capture the vision plane attention of a user in realtime? Eye gaze is a good clue to represent people's interest and intention [4,5,14,24,25], and smart glasses with the eye camera (*i.e.*, camera capturing images of the user's eye) enable realtime eye movement tracking. In this work we propose to address this problem by inferring the gaze direction of a user from his/her eye movements. The challenges in achieving efficient eye gaze detection lie in three aspects: the bulky hardware and power hungry tracker, the user experience issue and the scalability. Existing state-of-the-art eye gaze determination solutions [11, 18, 20] usually leverage two cameras to conduct 3D reconstruction, which incur high hardware, energy and computation cost for resource limited wearable glasses. A few work need only one camera [7, 9, 12], but they require personalized training or calibration. Instead, our approach is designed to achieve high accuracy with only one low-resolution camera. For better user experience, our solution should put no restriction on the user's movement, and the set up (*e.g.*, calibration) difficulty should be minimum. Besides, we need to reduce power consumption of continuous image capturing and processing. To support these, we develop a low-cost glasses hardware, with an eye camera, a microphone and speakers, which are embedded with our software for visual attention capture.

2) *How to match the target on device plane according to user's visual attention?* In our visual attention driven network, we face the issue of addressing a network device without any pre-exchanged contact information, and match a user's visual target with the target's device. At a high level, we address this problem based on the observation that the direction of user's visual attention (indicated by sightline direction) is consistent with the direction from the user to his/her target, and is also consistent with the direction between smart the glasses worn by them. Technically, we capture the direction of a user's sightline with the onboard sensors and camera. We also capture the direction from user's smart glasses to the target smart glasses by tracking the subtle relative displacement of two head-mounted speakers during arbitrary head movements. It is also a challenge to reduce the negative impact of inaccuracy of earth coordinate and device coordinate system on matching the device directions and eye gaze directions.

In summary, the **contributions** of this work are as follows:

1) painting the vision of a new networking mode in which connections are established following users' visual attention and introducing the first system that enables the user communicate and information exchange with surrounding world simply by a gaze.

2) presenting a novel scheme to capture a user's visual attention and then the visual target. We also propose a Phase Locked Loop (PLL) based solution to estimate the direction from the user to his/her target leveraging the Doppler effect caused by mild arbitrary head movements. Then we correlate the user's visual target to target smart glasses with a double-matching strategy.

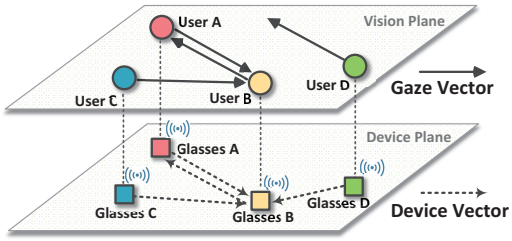


Figure 1: Gaze vectors and device vectors of a visual attention based networking system.

3) designing and implementing a platform, **iGaze**, for the visual attention driven networking, consisting of a set of hardware and software. Our system can accurately capture the user's vision direction and successfully connect the user's device to a target.

4) using the proposed system to conduct a series of trials for various application scenarios to demonstrate the performance of our system. Our extensive user studies show that our system works robustly across individuals. Our lab testing shows that 95% of users attentions are captured correctly by setting the fixation duration window to $0.6s$. The mean error of relative gaze direction computed is less than 5^o, and the error in the relative device direction is less than 5° when two users are separated by 4 meters.

2. SYSTEM OVERVIEW

2.1 iGaze Design Space

iGaze is designed to enable a new communication mode for smart glasses users. We focus on making the networking devices smarter that can understand the user's attention and automatically connect to the target of interest.

• **iGaze** has separated and modularized hardware and software design. It can run on top of existing networking protocols, *e.g.*, Wi-Fi.

• **iGaze** applies to scenarios where networking participators do not need to exchange contact information such as IP, email, web account before connecting to each other.

• **iGaze** assumes the initiator is equipped with camera to capture her visual attention at realtime and all participating devices possess computing, networking functionalities and some common sensors, *e.g.*, gyroscope and microphone.

• **iGaze** applies to scenarios where the visual target for communication is relatively static during the connection initialization to achieve high matching accuracy. One of our future works is to enable **iGaze** to locate and connect to mobile targets at a reasonably high moving speed.

Each user is associated with a *co-located* smart device. That is, the device is regarded as being at the same location with its user. In the world through a user' eyes (*vision plane* shown in Fig. 1), there are objects in the field of view and he/she usually gazes at visual targets of interest to him/her, *e.g.*, another user or an artwork. In the cyber world (*device plane* in Fig. 1), a device only "sees" the network identifiers of other devices in proximity, without knowing which one is corresponding to which target in vision plane. The gap between these two planes raises a challenge for our system.

Two types of scenarios are considered in this work: social networking and object-oriented augmented reality. In the first one, the user wears smart glasses and looks around while the glasses are tracking his/her gaze direction. When two users have an eye

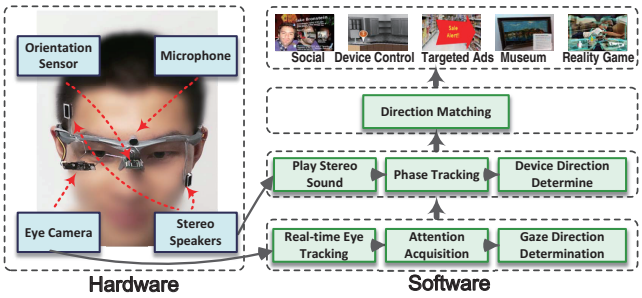
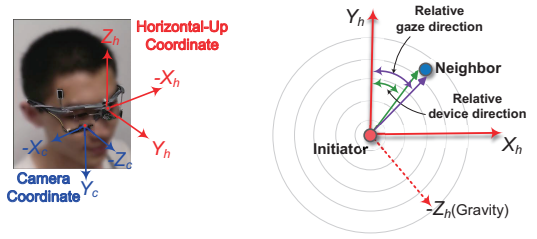


Figure 2: Our smart glasses iGaze prototype.



(a) The HU coordinate of our glasses.

(b) Relative gaze/device direction in the HU coordinate.

Figure 3: Direction representation in the HU coordinate system of our glasses.

contact, a network connection is built between their glasses. This scenario is fitted into many social applications. This type of applications require all participating devices to possess eye cameras, and we refer to this mode as *bidirectional* application mode. In the second case, the user wearing smart glasses wants to obtain certain information about the visual target object by gazing it. For example, visitors can obtain the description of an artwork when they gaze at it for a certain time period. In these applications, the target objects need to have devices (without cameras) to respond to the 'gaze query', and we refer to this mode as *unidirectional* application mode.

2.2 Principal of Gaze Based Networking

What we look for is a bond between the vision plane and device plane, which are dissociated currently. Note that finding the gaze direction and finding the direction of pairing devices are themselves challenging tasks. To address this association issue, we raise our idea based on the following observations. As shown in Fig. 1, when a user A is looking at a visual target B, we define the observer A's **gaze vector** as $\mathcal{V}_A$, whose direction is the direction of A's sightline and magnitude is the distance from A to B. In the device plane, we define the **device vector** $\mathcal{U}_{AB}$ as the vector from A's glasses to B's glasses. Both types of vectors can be denoted as a 2-tuple (α, ℓ), where α is the direction and ℓ is the magnitude. We define the concept of *consistency* between a gaze vector and a device vector.

DEFINITION 1. *A gaze vector (α_g, ℓ_g) is said to be* consistent *with a device (or another gaze) vector (α_d, ℓ_d) if their directions and magnitudes hold the following conditions: $|\alpha_g - \alpha_d| < \theta$ and $|\ell_g - \ell_d| < \eta$. Here θ and η are predetermined constants that will impact the false positive and false negative ratios of **iGaze** in building gaze based connections.We will use θ as the direction consistency threshold in the rest of the paper.*

Two observations help us to design our visual attention driven networking protocol.

OBSERVATION 1. *Given an observer and his/her gaze vector, among all device vectors started from the observer's device, only the device vector to the correct visual target's device is consistent with the gaze vector.*

As an example in Fig. 1, for the observer A, only the device vector $\mathcal{U}_{AB}$ is consistent with the gaze vector $\mathcal{V}_A$. Then in the unidirectional mode, our system learns that device B is the correct match to A's visual target. In the bidirectional mode, establishing a connection requires a pair of users to look at each other. Our system should also confirm that device A is the correct match to B's visual target, *i.e.* $\mathcal{U}_{BA}$ is also consistent with $\mathcal{V}_B$. Note that, $\mathcal{U}_{BA}$ and $\mathcal{U}_{AB}$ have same magnitude and opposite directions. For a practical bidirectional system, since smart glasses continuously capture the direction of the user's sightline, it is not necessary to determine all device vectors from device A to all its neighbors.

OBSERVATION 2. *Given a pair of users who are looking at each other, their gaze vectors have opposite directions.*

Taking the scenario in Fig. 1 as an example, both gaze vectors $\mathcal{V}_B$ and $\mathcal{V}_D$ have opposite direction to $\mathcal{V}_A$. Thus user A can firstly exclude user C by the direction of gaze vectors and only compute vectors $\mathcal{U}_{AB}$ and $\mathcal{U}_{AD}$ in the device plane. For bidirectional applications with many users, the first round of sightline direction matching can greatly reduce the overhead in determining pair-wise device vectors.

2.3 System Architecture

Our system includes both software components for visual attention driven networking and the smart glasses hardware.

Software: There are three major building blocks:

• **Gaze vector acquisition**: This component includes three modules. The real-time eye tracking module captures the movement of a user's eye using the eye camera at real-time. Attention acquisition component takes the eye movement data as input to detect a visual attention when the gaze lasts for a reasonable time. The threshold is predetermined and can be dynamically adjusted by users. After a visual attention has been captured, the gaze vector determination calculates the corresponding gaze vector to the visual target. To avoid unwanted paring when eyes happen to be fixed on a certain object, we ask the users to make a mild head gesture (*e.g.,* head nod) as the postfix of the attention.

• **Device vector estimation**: When the user makes a head gesture, his/her smart glasses emit inaudible acoustic signal to adjacent devices. Candidate devices invoke the phase tracking and device direction determination modules to estimate the device vector by exploiting Doppler effect.

• **Visual attention driven networking (VAN)**: It defines the message format and communication process in our visual attention driven networks. **VAN** constructs a connection from initiator to the visual target's device based on the matching results of the gaze vector and device vectors.

Hardware: In this work, we design our own hardware to support the system functionalities as shown in Fig. 2. Limited smart glasses are available on the market with high prices, and their software platforms are diverse. To explore the feasibility and efficiency of our proof-of-concept prototype and conduct multi-user networking experiments, we instrument **iGaze** glass, which is a low-cost head-mounted smart glass. It has an eye camera, an orientation sensor (including a gyroscope and a magnetic sensor), a microphone and two speakers. Using the gyroscope sensor, we can determine the pose of our glasses. The direction of gaze vector is captured by eye camera and the device vector is determined using two speakers, one microphone and the gyroscope.

In order to evaluate the consistency between the gaze vector and device vectors and reduce the negative impact of coordinate

conversion, we introduce a temporary coordinate system, named *horizontal-up coordinate* (HU coordinate), which can be determined using only gyroscope. For each round of connection construction, the gaze and device vectors are calculated in the same HU coordinate system. When an initiator's attention is captured, our system uses the pose of the initiator's glasses at this very moment to determine the HU coordinate, whose origin is the origin of his/her gyroscope. Its Z axis always points to the up direction (opposite to the gravity direction) and its X and Y axes lie on the horizontal plane. The X axis is the projection of the line passing through two speakers. Our system also records the current pose of initiator's gyroscope as its initial state. Using the eye camera, we can obtain the gaze direction in *camera coordinate* as shown in Fig. 3(a). As the frame of the glasses is rigid and the relative position between the gyroscope and eye camera is fixed, the gaze vector can be simply transformed from camera coordinate to HU coordinate with known fixed system parameters. The details will be discussed in Section 3. Also, the device vector is determined in HU coordinate using two speakers, one microphone and the gyroscope sensor (Section 4).

As shown in Fig. 3(b), based on Observation 1, we use the HU coordinate instead of the earth coordinate for accurate direction matching. The reasons are two-fold: (1) **iGaze** only needs the relative directions with respect to initiator to match gaze vector and device vector. (2) A gyroscope is enough to determine the device pose in HU coordinate. For the earth coordinate, a magnetic sensor is used to understand north and east, but the accuracy of the magnetic sensor is much lower than the gyroscope. Converting directions in HU coordinate into earth coordinate could cause an accuracy loss.

Note that in the bidirectional mode, based on Observation 2, a coarse matching between the initiator and neighbors is conducted using their gaze directions in the earth coordinate. For this purpose, the gaze directions in the earth coordinate are obtained using the magnetic sensor, and low accuracy (*e.g.*, $\pm 20^{\circ}$) gaze directions can sufficiently filter a large portion of unlikely candidates. But we still use direction in the HU coordinate for the final match decision.

2.4 Networking Protocol

While a user A is using our glasses, once he/she pays attention to a target, his/her gaze vector pointing to the visual target is obtained. Then the system runs the **VAN** protocol to build a network connection between A and his/her target. Here the user A plays the role of the *initiator* of this protocol. As illustrated in Fig. 4, in a *bidirectional mode*, a connection establishment requires two users to pay attention to each other, and the **VAN** protocol works as the following steps. (1) The initiating **iGaze** glass issues a connection request by broadcasting the direction α_r of his/her gaze vector to all neighbors. Neighboring devices can connect each other using Wi-Fi direct in an Ad Hoc mode, or communicate in the same subnet by joining the same access point (*e.g.*, "Starbucks" or "MobiCom 2014"). In other cases, devices can connect to a server. With a basic location service (*e.g.*, GPS), the server can determine the initiator's neighbors efficiently and transmit messages among them. (2) Every neighbor **iGaze** glass caches this request. Within a gaze direction matching time window, each **iGaze** glass waits for a local visual attention event (*i.e.*, the user is paying attention to an object). If there isn't any event before the expiration time, **VAN** just drops this request, otherwise it compares the direction α_r of the request gaze vector and direction α_l of local gaze vector. Based on the Observation 2, if $\pi - \Theta < |\alpha_r - \alpha_l| < \pi + \Theta$, **VAN** replies the initiator as a *candidate* of the target device and starts to capture acoustic signal using the microphone. Here Θ is larger than the direction consistency threshold θ, since the gaze direction matching uses in-

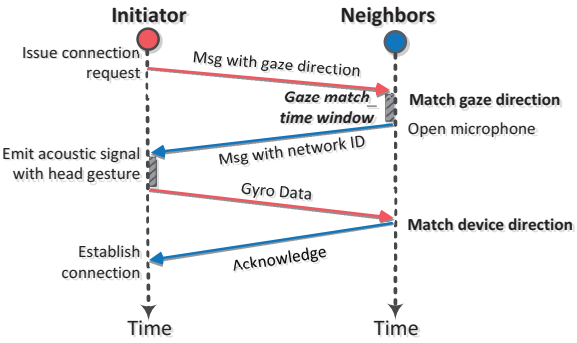


Figure 4: Networking protocol (VAN) after a gaze.

accurate directions in the earth coordinate, and it only aims to filter out noncandidates. (3) The initiator waits for replies until the expiration time. If there are any replies within the time window, the initiator makes a head gesture as an acknowledgement, and meanwhile, his/her **iGaze** glass emits a stereo acoustic signal. (4) Each candidate device receives the signal and calculates the direction β of the device vector from the initiator to itself. Based on the Observation 1, if $|\alpha_r - \beta| < \theta$, then it is the target device and replies an acknowledgement to the initiator. (5) Based on the acknowledgement, the initiator maps his/her gaze vector to the network ID of the target. Then the initiator can obtain information from the target or start a communication between them.

In the unidirectional mode, **VAN** runs similarly as in the bidirectional mode, except the Step (2). In Step (2), each neighbor doesn't wait for a local visual attention event or compare the gaze directions. As soon as it receives the request, it directly replies the initiator as a *candidate* of the target device and starts to capture the acoustic signal.

3. GAZE DIRECTION ACQUISITION

With the eye camera, **iGaze** tracks the iris of the eye and determines the gaze vector. Many sophisticated eye gaze determination solutions, *e.g.* [11, 18, 20], use two cameras for 3D reconstruction to achieve high accuracy. But those methods incur high hardware, energy and computation cost for resource limited wearable glasses. There are a few works that need only one camera [7, 9, 12, 29], but they require personalized training or calibration. To overcome those limitations which may hinder the adoption of gaze tracking techniques for wearable glasses, we design our head-mounted hardware to possess a gyroscope which has fixed relative position to the eye camera. With the hardware, our gaze tracking approach has the following advantages:

• Our glasses achieve sufficient gaze direction accuracy for applications using only one low-resolution camera. Simple geometry model and carefully designed methods are applied to remove the ambiguities caused by single eye image, also to reduce the computation complexity.

• For good user experience, our solution has no restriction on the user's movement. The paring protocol only needs relative gaze direction in the temporary HU coordinate of the glasses, which can be obtained using eye camera and gyroscope. As a result, little manual set up (e.g., calibration) is required for user's convenience.

• Instead of tracking eye movement continuously, we use the gyroscope to detect a head fixation before invoking the camera to capture and process the eye image, which greatly reduces the energy and computation cost. We also detect eye fixation before gaze direction calculation, which saves computation too.

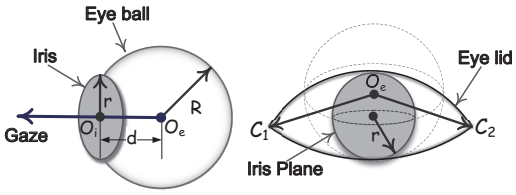


Figure 5: A simplified geometry model of eye.

3.1 Geometry Model for Gaze Detection

Eye model. The structure of eye can be closely approximated using a simple geometry model as presented in Fig. 5. The eyeball is a sphere whose radius is R. The contour of the iris plane is a circle of radius r. The gaze direction can be determined by the optical axis of the eye, which passes through the center of the eye ball O_e and the center of the iris O_i. While the user is looking around, the eyeball rotates around its center O_e and the gaze direction changes.

Projection model. By detecting the iris in the image captured by the eye camera, we get the projection of the iris on the image plane. The projection model is presented in Fig. 6. There are three coordinate systems involved. The origin of the *eye coordinate* is the center of the eye ball O_e. Its X and Y axes lie on the iris plane and the X axis is the intersection line of the iris plane and the horizontal plane. Its Z axis is from the eye ball center O_e to the iris center O_i, *i.e.* the gaze direction. The *camera coordinate* has the principal optical axis as its Z axis. Its X and Y axes are aligned with the axes of the camera's image plane. As presented in Section 2.3, the gaze vector is firstly obtained in camera coordinate and then converted into HU coordinate for comparing with device vectors.

Basic idea. The iris contour is a circle, but its projection on the image plane is elliptical. Intuitively, when the eye looks straight ahead, the projection of iris looks more like a circle; when the eye looks off to one side, it looks more and more close to an ellipse. When the relative position between the eye and camera is fixed, the ellipse parameters are determined by the norm direction or the iris plane, *i.e.* the gaze direction. Based on this idea, we can estimate the pose of the iris circle by back-projecting the ellipse onto a circle in 3D space. The challenge is that with only a single elliptical image, there are many circles satisfying the projection cone. In this work, we remove ambiguities using anthropometric property of eyeball, the hardware position and the principal in [29]. Based on our design, the system requires little manual calibration. Moreover, to reduce computation, we acquire the user's attention before calculating his/her gaze direction. And we further reduce the cost for continuous image capturing and processing greatly by detecting head fixation using gyroscope before opening the camera module.

3.2 Visual Attention Detection

Before calculating the gaze direction, we first detect the user's attention. Many results in the cognitive and neural areas show that attentional and oculomotor processes are tightly integrated at neural level and there is a close relationship between visual attention and eye fixation [4,5,14,24,25]. There are two basic statuses of eye-movement: fixations and saccades. Considering the *fixation* status as the sign of visual attention, we use an eye-movement velocity-threshold method to detect the fixation for its better efficiency over other similar methods [5, 24, 25]. First, we binarize the image captured by the eye camera with a threshold, and detect the iris area based on its contour. Note that the threshold is calculated by an adaptive algorithm which provides proper thresholds for different eye colors and light conditions. Then, we fit the iris area into an ellipse eye model as in Fig. 6. By detecting the real-time position (I_x, I_y) (Fig. 6) of the iris ellipse, our method calculates the point-to-point eye-movement velocity of two continuous iris images. Within a time window, if the mean of velocities is lower than a given threshold, then it is a fixation, otherwise it is a saccade. Unlike gestures who typically have a prefix (e.g., "OK Glass" of Google Glass) to avoid false alarm rate, gaze lacks such prefix. To avoid false positive detection of the attention, we introduce the head gesture as the user's postfix of his attention, which is also used to detect the device direction (in Section 4).

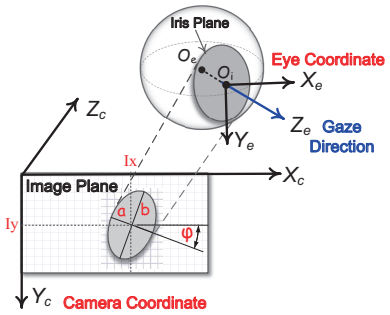


Figure 6: Iris projection on the image plane.

3.3 Direction of Gaze Vector

As soon as a fixation is detected, **VAN** is started to connect the target. At this moment, the HU coordinate system for this round is determined according to the head pose, and the pose of the gyroscope in this HU coordinate is also obtained. Then the iris ellipse of the fixation is used to compute the gaze direction. Let the transformation of the gaze vector from the HU coordinates $\mathcal{V}_h = (x_h, y_h, z_h)$ to the camera coordinates $\mathcal{V}_c = (x_c, y_c, z_c)$ by $\mathcal{V}_c = R_{hc} \cdot \mathcal{V}_h$. In our **iGaze** glass, the relative position between eye camera and the gyroscope is fixed (which is known as a system internal parameter), and the pose of the gyroscope in HU coordinate is known. Hence it is easy to determine R_{hc}.

Now the core issue is to determine gaze direction, *i.e.* the normal direction of the iris plane, in the camera coordinate system from a single elliptical image of the iris. We solve this problem in the following steps. Referring to the projection model in Fig. 6, the projected ellipse in the camera coordinate system is defined by the conic: $\mathbf{v}_c^T A \mathbf{v}_c = 0$. Here A is the conic matrix obtained by applying an affine transformation S to a conic H in the form $A = S^T H S$. Here S and H are determined by the ellipse parameters (Fig. 5) as

$$S = \begin{pmatrix} \cos\phi & -\sin\phi & -I_x\cos\phi + I_y\sin\phi \\ \sin\phi & \cos\phi & -I_x\sin\phi - I_y\cos\phi \\ 0 & 0 & 1 \end{pmatrix},$$

$$H = \begin{pmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

We define the transformation between the eye coordinates and camera coordinates as $\mathbf{v}_c = R_{ec}\mathbf{v}_e + \mathbf{t}$. Here $R_{ec} = (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ is a 3×3 matrix. Given a point in the iris plane $\mathbf{v}_e = (x_e, y_e, 0)^T$, let $\mathbf{u}_e = (x_e, y_e, 1)^T$ be the homogenous coordinate of $\mathbf{v}_e$. With $\mathbf{u}_c = \mathbf{v}_c/z_c = (x_c/z_c, y_c/z_c, 1)^T$ and $G = (\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{t})$, we have

$$z_c\mathbf{u}_c = G\mathbf{u}_e. \tag{1}$$

Using the eyeball model, the back-projected circle in the iris plane is defined by

$$\mathbf{u}_e^T Q \mathbf{u}_e = 0, \text{where } Q = \begin{pmatrix} \frac{1}{r^2} & 0 & 0 \\ 0 & \frac{1}{r^2} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

And its projection in image plane should be

$$\mathbf{u}_c^T A \mathbf{u}_c = 0. \quad (3)$$

Combing (1) and (3) yields

$$\mathbf{u}_e^T G^T A G \mathbf{u}_e = 0. \quad (4)$$

Because Eq. (2) and (4) define the same conic, we obtain

$$G^T A G = kQ, \quad (5)$$

here k is an unknown scale factor. Until now, we get the Eq. (5) providing 6 constraints. But there are 8 independent unknowns, three in R_{ec}, three in $\mathbf{t}$, k and r. We are interested in R_{ec} which determines the relation between gaze direction and camera orientation. Since the radius of the iris and eyeball are very close to an anatomical constants (7mm and 13mm) [18], we set $r = 7$ mm to reduce one unknown. Then by solving Eq. (5), we get eight solutions. We impose that, the gaze direction always pointing the front of the head and the iris must lie in front of the image plane, So, six solutions can be eliminated, leaving two solutions that cannot be disambiguated. Call the two solutions as $(\mathcal{V}_1, O_{i1})$ and $(\mathcal{V}_2, O_{i2})$, each solution consists of the gaze vector and the center of the iris. To eliminate the other incorrect solutions, we apply the principal in [29], that the distance between the two corners of an eye and the center of the eyeball O_e should be equal. Hence, as shown in Fig. 5, $O_eC_1 = O_eC_2$ must be satisfied. Also we have $d = \sqrt{(R^2 - r^2)} = 11mm$. With each solution, we get the center of the eye ball O_e in 3D space, and compute a pair of O_eC_1 and O_eC_2. If $|O_{e1}C_1 - O_{e1}C_2| < |O_{e2}C_1 - O_{e2}C_2|$, then $(\mathcal{V}_1, O_{i1})$ is chosen as the solution, otherwise $(\mathcal{V}_2, O_{i2})$ is the solution.

With this technique, we achieve sufficient gaze direction accuracy using one eye camera with low computation cost. In our system implementation, we further reduce the computation and energy overhead by detecting a head fixation using gyroscope before image capturing and processing, and we will discuss the detail in Section 5.

4. DEVICES DIRECTION ACQUISITION

In **VAN**, as shown in Fig. 4, after the gaze direction matching phase, the initiator needs to determine the direction of the device. There are a few methods proposed to estimate device direction by measuring relative displacements [22, 28]. However, those efforts target handhold smart devices, *e.g.*, smart phones. The ranging based methods usually require large phone displacement, *e.g.*, larger than 20cm [22], and the Doppler effect based methods require high movement speed, *e.g.*, $2 \sim 6m/s$ [28], both of which are hard to achieve by head-mounted glasses. For smart glasses, the new challenge is to precisely estimate device direction with arbitrary mild (small amplitude and speed) head movements which make the Doppler effect very subtle to measure.

Facing this challenge, we propose to track the subtle relative displacement of two head-mounted speakers, *i.e.*, variants of distance between the speakers and the receiver, based on the Phase Locked Loop (PLL) technique. Specifically, when the initiator makes a head gesture, *e.g.*, head nod, two speakers of our glasses broadcast a binaural signal with two sine waves at different frequencies, which are at high-frequency channels and inaudible to human. When a target device receives these signals, the frequency of each signal is shifted due to the Doppler effects. By PLL, the receiver can track the precise phase of the received signal, where the phase shift is in proportion to relative displacement. In this way, the measuring accuracy of the relative displacement is less than $1mm$, with a small distance between the two speakers, *i.e.*, $18cm$. Finally, we obtain the relative direction of the receiver to the initiator by studying its relation with the relative displacement. With our method, any form of head gestures are supported. Considering that too complicated gestures may impose inconvenience for users, our system provides two simple gestures, head nod and shake, as the default setting.

In the rest of this section, we present details of our techniques for relative displacement tracking and direction measuring.

4.1 Tracking Relative Displacement

Assume the source broadcasts non-audible sine wave at frequency f_a. Meanwhile, the source is moving towards the receiver at velocity v while the receiver is static. Then, the frequency f_r of received sine wave is shifted due to Doppler effects. Specifically, $f_r = \frac{v_a}{v_a - v} f_a$, where v_a is traveling speed of sound. Hence, the frequency shift $f = f_r - f_a = \frac{v}{v_a - v} f_a$. In our case, since $v \ll v_a$, $f \approx \frac{f_a}{v_a} v$.

We then deduce the relationship between phase shift and relative displacement (from the acoustic source to the receiver). Assume the emitted signal is $s(t) = \sin(2\pi f_a t)$. Due to Doppler effects, we model the received signal as

$$r(t) = \sin(2\pi f_a t + \phi_t) \quad (6)$$

Here, ϕ_t is the phase of the received signal with $\frac{d\phi_t}{dt} = 2\pi f(t) = 2\pi \frac{f_a}{v_a} v(t)$. Hence, when the source moves from place A_1 to place A_2 (with $\|A_1A_2\| \leq$ half of wavelength) and the receiver is at place R, the relative displacement is

$$d = \|RA_1\| - \|RA_2\| = \frac{v_a}{2\pi f_a}(\phi_{t_1} - \phi_{t_2}) \quad (7)$$

where ϕ_{t_1} and ϕ_{t_2} denote the phase of the received signal when the source is at place A_1 and A_2 respectively. Therefore, the relative displacement can be inferred from the calculated phase which is further detailed in the next subsection.

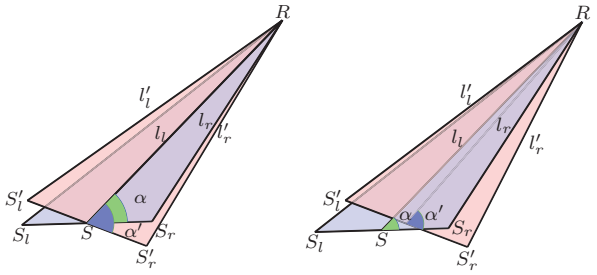
4.2 Tracking the Phase of Received Signal

We calculate the real-time phase ϕ_t from the received signal. To track the phase from $r(t)$, the received signal needs to be preprocessed. More specifically, the actual received signal before preprocessing is denoted as $r_0(t) = A_t \sin(2\pi f_a t + \phi_t) + \sigma_t$, where σ_t is the noise and A_t is the changing amplitude. The original signal $r_0(t)$ first passes through the Band Pass Filter (BPF) to remove the noise σ_t. Then, the filtered signal reshapes its wave by using Automatic Gain Control (AGC) [23]. After that, the changing amplitude A_t becomes approximately constant 1, which means A_t is also eliminated and the processed signal is then close to $r(t)$. Finally, we use Phase Locked Loop (PLL) [1] to calculate the real-time phase ϕ_t from $r(t)$ in Eq. (6). To get the precise ϕ_t, we update an adaptive estimation of ϕ_t in real time, denoted as $\theta[k]$, to achieve that $\theta[k] \approx \phi_t$, where $t = kT_s$ and T_s is the sampling period of audio recording. To make θ converge to ϕ after enough iterations, we define the corresponding function $J_{PLL}(\theta)$ such that J_{PLL} converges to its maximum at the same time when θ converges to ϕ. We update $\theta[k]$ in the iterations as $\theta[k+1] = \theta[k] + \mu \frac{dJ_{PLL}}{d\theta}$, where μ is a small positive constant that determines the speed of convergence in the iteration. According to [1], we choose $J_{PLL}(\theta) = \text{LPF}\{r(t)\cos(2\pi f_a t + \theta(t))\}$. Here, LPF is the Low Pass Filter which excludes high frequency component. The estimation of $\theta[k]$ is $\theta[k+1] = \theta[k] - \mu\text{LPF}\{r[k]\sin(2\pi f_a k T_s + \theta[k])\}$.

4.3 Measuring Device Direction

We then show how to measure direction in HU coordinate from initiator S to receiver R using relative displacement.

Notice that an **iGaze** glass S has two speakers, denoted as S_l and S_r. Denote the distance from R to S_l and S_r as l_l and l_r respectively. Hence, l_l and l_r change continually when the device



(a) Case 1. (b) Case 2.

Figure 7: Computing direction from device S to device R

S is moving. In other words, there are relative displacements $d_l = l_l - l'_l$ and $d_r = l_r - l'_r$ and they can be tracked in real time. Furthermore, the relative displacements are then used to calculate the direction by combining the tracked relative displacement and the real-time direction of S_lS_r. The other input is the distance of the two speakers $\varpi = |S_lS_r|$. Therefore, $|S_lS| = |SS_r| = \frac{\varpi}{2}$. We also let S be the midpoint of line segment S_lS_r. Assume that the direction of S_lS_r and SR at the projection of horizontal plane in WCS is β and γ. Then γ is the target angle we calculate. Assume $\hat{\alpha}$ is the projection of α at the horizontal plane. Since $\gamma = \beta + \hat{\alpha}$ and β can be obtained from inertial sensors of the mobile device (*i.e.*, gyroscope, accelerometer), we can calculate the angle α from which γ is indirectly obtained. To illustrate our method in detail, we first consider a simple case (case 1) and then a more general case (case 2) as follows.

Case 1: Two speakers rotate around the midpoint S of two speakers in 3D space, shown in Fig. 7(a).

Hence, the distance $l = |SR|$ does not change when rotating. Actually in this case, only one speaker is needed for calculating α. For instance, we use the relative displacement of speaker at S_2. According to the law of cosines,

$$\begin{cases} l_r^2 = l^2 + \frac{\varpi^2}{4} - \varpi l \cos\alpha \\ l_r'^2 = l^2 + \frac{\varpi^2}{4} - \varpi l \cos\alpha' \end{cases} \tag{8}$$

In our case, l is from 2m to 8m and $|S_lS_r|/2 < 0.09m$ that l is much larger than $|S_lS_r|/2$. Moreover, in our experiment, we use the calculated displacement d_r at the time interval of 0.01s that $|d_r| = |l_r - l'_r| < 0.01s * 1m/s = 0.01m$. Hence, it can be inferred that $l_r \approx l'_r \approx l$.

From Eq. (8), we have $\varpi\cos\alpha' - \varpi\cos\alpha - 2d_r \approx 0$. Note that α' can be calculated by gyroscope if α is given. Hence, for the series of tracked d_r and gyroscope data, α can be estimated from this equation by Maximum Likelihood Estimation.

Case 2: Two speakers rotate arbitrarily around any point in 3D space, as in Fig. 7(b). The point it rotates around may change arbitrarily when moving.

A key observation here is that the movement of **iGaze** glass can be decomposed into two movements: rotation around midpoint and translational movement, and the relative displacement caused by rotating component is very small (*i.e.*, less than $\frac{18}{2}$cm). Consequently, the displacement component caused by translational movement becomes relatively high. To address the challenge raised by possible arbitrary head movement, we utilize the relative displacements of two speakers (*i.e.*, d_l, d_r), and $d_l - d_r$ eliminates the displacement resulting from translational movement. Hence, we can infer an equation

$$\varpi\cos\alpha' - \varpi\cos\alpha + d_l - d_r \approx 0 \tag{9}$$

As a result, similar to Case 1, we estimate the angle α using this equation and finally obtain the device direction γ.

Component	Description
Raspberry Pi II	700MHz CPU, 512MB RAM, Linux
Eye Camera	Maximum Resolution 640x480 @ 30fps
Gyroscope	MPU6050, 6-axis, Accuracy - 0.01°
Magnetic Sensor	3-axis, Accuracy - $512c/G$
Wi-Fi	OURLINK, 300Mbps, USB port
Microphone	Sample rate 44100Hz
Stereo Speakers	Frequency range 180Hz-20KHz

Table 1: Specifications of iGaze glass

5. PROTOTYPE IMPLEMENTATION

Establishing network connection based on visual attention is an unexplored mode of communication. We design and implement a proof-of-concept smart glasses prototype system, named **iGaze**, to start network connection with the visual target. **iGaze** consists of a low-cost smart glasses hardware and a set of software. The system architecture is illustrated in Fig. 2. After we address the challenges of determining gaze direction in Section 3 and device direction in Section 4, here we present the hardware and software design issues and the implementation specifications.

5.1 Hardware Prototype

Fig. 2 illustrates our prototype hardware. To obtain gaze direction, **iGaze** glass has a fixed eye camera tracking the eye movement. It is 3cm away from the eye. A fixed gyroscope is embedded in the front-head position of the glasses to determine device pose in the HU coordinate and simplify the coordinate transformation. A magnetic sensor is co-located with the gyroscope, which is only used in the bidirectional mode to exclude unlikely neighbors with gaze direction. For the devices direction, **iGaze** glass has a central microphone and two separated speakers with a fixed distance (18cm).

A platform is required for the data processing, which should be light-weight and convenient for various **iGaze** applications development. In our implementation, we use Raspberry Pi with a Wi-Fi module. It is a credit-card-sized single-board computer with Linux operating system. All head-mounted sensors are connected to its hardware data interfaces. It supports a wide range of programming languages, *e.g.* Python, C++, Java and Perl. Our prototype system is mainly designed to explore the feasibility of visual attention based network connection establishment. And the current version of our glasses doesn't have a head-mounted screen. The data received and processed by Raspberry Pi can be accessed by any smart device with a screen (*e.g.*, smart phone, pad and laptop) via wireless channel in real time. The specifications of **iGaze** glass is presented in Table 1.

5.2 Software Prototype and Key Parameters

We implement all software blocks of **iGaze** for Linux using C++. It supports both unidirectional and bidirectional application modes. In the unidirectional mode, the target could also be a device without an eye camera. So we also implement the device direction estimation and **VAN** for Android system using Java, which enables our glasses to connect an android device in unidirectional applications. The visual attention acquisition components are developed based on the OpenCV library.

Specifically, the eye camera captures images at the frequency of 30fps with a low resolution 320×240. To identify visual attention, the eye movement velocity threshold is set to 20pixes/s in the scale of the eye image and the detection time window is set to 0.6s. We will further analyze these two parameters in Section 6. In the unidirectional mode, the initiator nods head after he/she pay attention to a target as a postfix of attention. But for the bidirectional mode, there is a gaze direction matching phase. Once there are replies

from coarsely matched users, our system uses a beep to inform the initiator to nod his/her head. While the initiator nods head, the speakers play a binaural acoustic signal consisting of two inaudible high frequency sine waves. In the default setting, the left track is 19KHz and the right track is 19.5KHz. For the wireless communication of **VAN**, both AP and Ad Hoc modes are implemented.

In our implementation, we improve our software to address the challenge that, the smart glasses are resource limited, while image processing could be computation-intensive and power-hungry. We use a trick to reduce the overhead in real-time eye-tracking, while keeping the attention acquisition accuracy. We observe the data of orientation sensor of 30 volunteers and notice that, when a user is paying attention to a visual target, his/her head is usually in a fixation status. Specifically, the orientation changes around three axes (especially the Z axis) show plateaus during the staring. Analyzing orientation data (sample rate is 50Hz) is much more efficient than processing the video. So, in our system, eye tracking only starts when the variance of three orientations are within a threshold for a small time window. In our experiments, the time window is 10 samples (0.2s). With this improvement, significant computation and energy cost for image processing can be saved.

There is also another implementation issue that needs to be addressed: when a receiver computes his/her direction to the initiator, the acoustic wave is recorded by the receiver while the corresponding glasses orientation samples are from the initiator's **iGaze** glass. Before calculating the direction, the acoustic wave and orientation samples need fine-grained synchronization. As **iGaze** glass records orientation sensor samples exactly when it is broadcasting the acoustic signal, so the challenge is to locate the start time of the initiator's sine signal in the recorded noisy wave. We observe the sequence of relative displacements of two speakers changes smoothly if they are calculated using the expected sine signal, otherwise, it jitters severely. To achieve coarse-grained synchronization, we divide the displacement sequence into segments and calculate the standard deviation (std) of each segment and find the segment from which the std falls below a threshold, *i.e.*, the start time of the initiator's sine signal. Specifically, each segment contains 500 consecutive calculated displacements (about 0.01s), and the threshold is set to 0.2cm. Then we refine the synchronization by adding the synchronization error as an unknown to the Maximum likelihood Estimation of the device direction. That is, we make Maximum likelihood Estimation according to Eq. (9) with 2 unknowns: the synchronization error and the pair-wise direction. In our experiment, we find that the coarse-grained synchronization achieves a very small error (within 0.15s). And the refinement further reduces the error with little overhead due to small searching space.

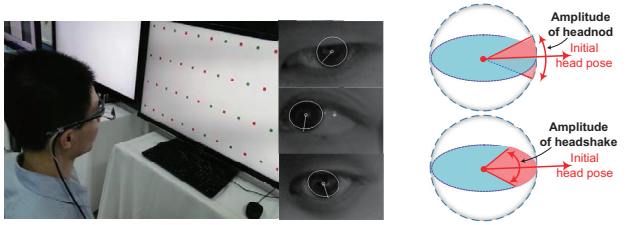
We develop two applications for system evaluation: (1) a social application in bidirectional mode, by which people gaze at each other to express their intention for friending. Once a pair of users are matched successfully, they follow each other on Twitter automatically. (2) a smart device application in unidirectional mode, by which a user looks at a sensor among various sensors placed on desks, a connection between them is built and the real-time sensing data is sent to the user.

6. EXPERIMENTS AND EVALUATION

We evaluate **iGaze** by measuring the micro-benchmark performance and studying two typical applications.

6.1 Micro Benchmark

We investigate the following metrics: accuracy of gaze direction and device direction, and power consumption.

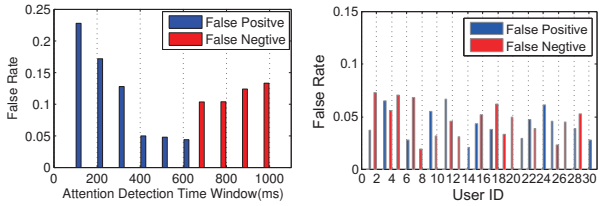


(a) Gaze capture. (b) Head gestures amplitude.

Figure 8: Experiment and samples of gaze capture, and amplitude of two basic head gestures.

6.1.1 Gaze Direction Accuracy

In this part, we evaluate the accuracy of our attention acquisition and relative gaze direction (as shown in Fig. 3(b)) estimation methods with the help of a large screen (47-inch). The screen displays marks which are separated from each other with a fixed distance. We have 30 volunteers (12 female and 18 male) wearing **iGaze** to pay attention to these marks in sequence, while **iGaze** is capturing their eye movements. Fig. 8(a) shows the experiments scenario and samples of our eye tracking results. Note that, the gaze direction accuracy is equivalent to the angle resolution of **iGaze**. So the distance to the screen won't effect the accuracy measurement. Before each run, the relative position between **iGaze** and the screen is recorded and the volunteer changes his/her attention by rotating eye ball. For each volunteer, about 100 gaze data are collected.

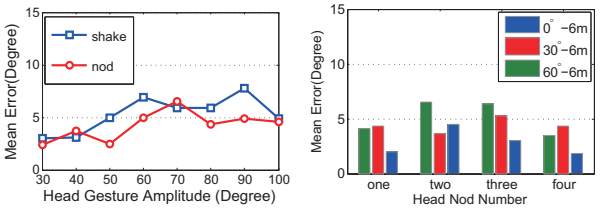


(a) False rate vs time window. (b) False rate vs users.

Figure 9: False positive and negative of attention detection against different time windows and users.

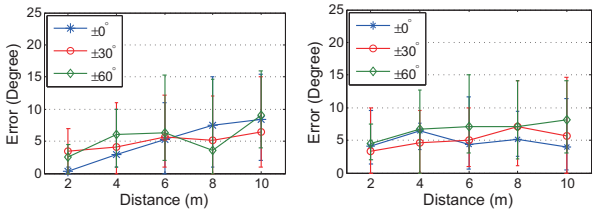
To get the true attention, we manually label their eye tracking data by asking volunteers to hit the blank key when they are paying attention to a marker. Two factors determine the accuracy of attention detection: attention detection time window and the eye movement velocity threshold. Fig. 9(a) presents the relationship between the time window and false rate. As the time window is getting larger, the false positive reduces while false negative increases. In the following experiments, we choose the window size when the two false metrics are equal (0.6s in our experiments). Obviously, larger threshold will cause higher false positive. However, the threshold setting could be tricky due to anatomical diversity. As shown in Fig.9(b), when we set the threshold to 20 pixels/second for our glasses, the accuracy varies among users. In the worst case, the false negative is 7% for the user No.2. But overall the correct attention detection ratio is $\simeq 95\%$.

It is difficult to obtain the true gaze direction of people. As a result, we test the equivalent angle resolution of **iGaze** by adjusting the density of markers on the screen, and estimating the marker the volunteer is paying attention to. We get the error range of calculated gaze direction by comparing the estimated markers with the true sequence. The highest resolution we test is 5°, *i.e.*, from the volunteer's view these markers are 5° separated from each other. And in this case, the ratio of correctly estimated markers are about 91%. About 8% estimated markers are incorrect, but they are ad-



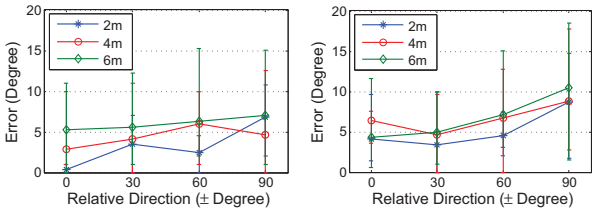
(a) Gesture amplitude. (b) Gesture number.

Figure 10: Device direction error v.s. head gesture patterns.



(a) Head nod. (b) Head shake.

Figure 11: Device direction error v.s. distance.



(a) Head nod. (b) Head shake.

Figure 12: Device direction error v.s. relative angle.

jacent to the correct markers. It means that, the error of the gaze direction is within 5° for 91% cases, and less than 10° for other 8% cases. For the rest 1% cases, the largest deviation is 15°. The current version of **iGaze** has a lower accuracy than some sophisticated solutions [11, 18, 20]. For example, a recent work use two camera for 3D reconstruction [11] to achieve 1° accuracy. But energy, computation cost and user experience are crucial factors for practical wearable devices, and image capturing and processing are especially power-hungry and computation-intensive. As a tradeoff, **iGaze** achieves sufficient accuracy for most social and augment reality applications with only one camera and requiring little manual calibration. With our modularization design, any future advance in gaze direction estimation can be adopted as the building block of iGaze.

6.1.2 Device Direction Accuracy

The speaker from the initiator emits a stereo acoustic signal while he/she makes a head gesture for a receiver to estimate their relative direction. Our system provides two default simple head gestures for users, head nod and shake. As illustrated in Fig. 8(b), the *amplitude of head gesture* defines the maximum angle to what extent the user nods/shakes his/her head. The *relative direction* defines the angle between the direction from *sender* to *receiver* and the orientation that the sender is facing in the HU coordinate, as shown in Fig. 3(b). We conduct a set of experiments in a $200m^2$ hall in an office building to measure the estimation accuracy. We have 30 volunteers to act as the initiator and nod/shake his/her head freely. One receiver records the signal at different distances and relative directions to the initiator. For each relative position, the experiment is repeated for 10 runs and these positions are manually marked as ground truth. First we investigate an interesting and important question that how sensitive is the estimation to the head gesture pattern of the initiator. Here, for the pattern, we refer to the gesture's amplitude and number. These factors affect the initiator's user experience since it is uncomfortable for user to move head drastically.

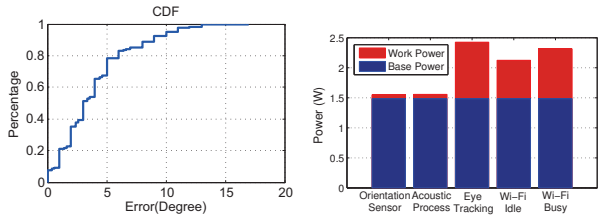
Pattern of head gesture. We test the impact of different head gesture patterns on the device direction accuracy under several fixed combinations of the device distance and relative direction. In the first test, the head gesture number for one round is fixed to two, and each volunteer runs the test with 20 times free head nod and 20 times free head shake. The amplitudes are calculated by head mounted gyroscope. We find diverse amplitudes among different users. Fig. 10(a) presents the mean direction error against different amplitudes for both nod and shake. The result shows that, fixing other variables, the direction estimation is not sensitive to the amplitude, and high accuracy (error $\leq 4^\circ$) can be achieved with only a mild nod/shake ($\leq 30^\circ$). In the second test, volunteers are asked to nod/shake their heads for different number of times. Due to user experience, we only evaluate the head nod/shake number up to 4. The results are depicted in Fig. 10(b). The estimation accuracy does not show an accuracy improvement with increasing head nod number and it is the same case for the head shake gesture. The reasons are two-fold: first, one head nod/shake has provided sufficient information to achieve high direction accuracy; second, the gyro data has a drift increasing with time which could cancel the benefit of more samples or even reduce the accuracy. In conclusion, our device direction estimation is robust against different head gestures and patterns and only one mild head gesture is sufficient for highly accurate estimation.

With free head gesture patterns, two factors affect the accuracy: distance and relative direction between devices. We evaluate one factor at a time while fixing the other.

Distance between devices:

Experiments are conducted to evaluate the impact of device distance on the direction accuracy with fixed relative device direction. At each distance, experiments are repeated 20 times for each head gesture. The results are plotted in Fig. 11 and show that for both head nod and shake gestures, our method achieves less than 5° error within 4 meters and achieves less than 9° error within 10 meters. It outperforms [22] whose accuracy is 10° with a 1m distance and [28] whose accuracy is 30° with a 4m distance. It implies that we can distinguish two targets ($4m$ away from initiator) whom are separated by $30cm$. As expected, when the distance gets larger, the accuracy decreases. But the error is still less than 15° in the worst case when the relative direction is less than $\pm 90^\circ$. The increasing error is caused by the multi-path effect for indoor environment and signal energy reduction.

Relative direction between devices: Then we fix the device distance to evaluate the relationship between direction accuracy and the receiver's relative direction to the sender. The results are plotted in Fig. 12. We find that, as the relative direction increases, there is higher chance that the receiver suffers a Non-Line of Sight (NLoS) acoustic signal. The NLoS effect reduces accuracy of measured relative displacement and the latter device direction. For the head shake gesture, the mean error increases from 6° to 10° when the relative direction changes from 0° to $\pm 90^\circ$. Head nodding suffers less from the NLoS effect, and its mean error remains less than 7° within the $\pm 90^\circ$ range. However, when the receiver moves to the back of the initiator, the accuracy declines sharply and the estimation tends to look like random since no emitted acoustic signal can directly arrive at the receiver. To remove potential mismatches caused by this effect, we notice that, when the minimum



(a) device direction error (b) Energy consumption

Figure 13: Device direction error and energy consumption of each module.

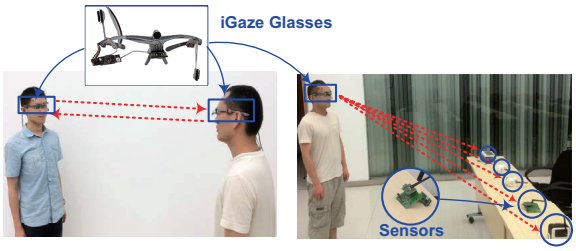


Figure 14: Experiment scenarios of two cases: social application (left), smart device application (right).

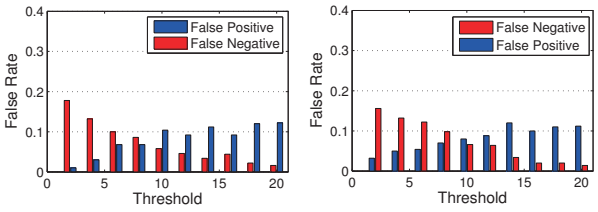
value, which is the result of Maximum Likelihood Estimation using Eq. (9), is higher than a threshold, it infers that the calculated device displacement is noisy and the estimated direction is not accurate. So **iGaze** sets a threshold on the minimum value and a receiver can find that his/her position is at the back of the initiator.

By placing the receivers uniformly in the initiator's field of view within range of 10m, we measure the overall accuracy of the device direction estimation. The initiator can nod or shake head as he/she likes. With 30 volunteers, there are 300 runs in total, and Fig. 13(a) plots the result. For 80% cases, the device direction error is less than 5° and for 95% cases, the device direction error is less than 10°.

6.1.3 *Computing Cost and Energy Consumption*

We evaluate the computation cost (delay) by measuring the runtime of each component. It takes the Raspberry Pi about 30 ms to process one eye image to obtain the gaze direction, including iris ellipse fitting and gaze vector calculation. To process the one-second acoustic signal and gyro data to get the device direction, the Raspberry Pi needs about 200 ms. Other computations (*e.g.*, coordinate conversion and direction matching) are negligible. As a conclusion, all computing of our system are completed in realtime.

We evaluate the energy cost of **iGaze** by measuring the current and voltage using oscilloscope. To test the power consumption of each component, firstly we test the base power of our glasses when all sensors and communication modules are off and no software component is running. Then, we run a single component each time while other components are off and test its power consumption. Fig. 13(b) plots measurement results. As expected, the orientation sensor (including a gyroscope and a magnetic sensor) and our device direction estimation component are quite power-saving, and the eye tracking component costs much more energy than them. Hence, as presented in Section 5, we improve the implementation of **iGaze** to greatly reduce the energy waste for unwanted image processing by filtering out saccade periods according to the real-time gyroscope status. With a normal 2600mAh lithium battery, iGaze can work continuously for about 5 hours when Wi-Fi module is always on, and eye camera is on for half the time (*i.e.*, the user's head is fixed for half the time).



(a) Social case. (b) Smart device case.

Figure 15: False positive and false negative changes with consistency threshold for two cases.

6.2 Case Based System Evaluation

Understanding the performance of each component, we then evaluate the integrated system by case studies. Two cases are considered: *social case* and *smart device case*. In the social case, we have 4 volunteers wearing **iGaze**, two of them pay attention to each other to build a networking. They change their relative position freely in a $100m^2$ room to evaluate the system performance. In the smart device case, 8 environment sensors are placed 50 cm apart from each other. 30 volunteers wearing **iGaze** pay attention to a sensor to "read" its real-time sensing data in the unidirectional mode. Fig.14 shows the experiment scenarios of two cases.

We are interested in the following metrics: communication cost, delay and connection correctness. For the social networking establishment, communication cost is mainly caused by transmitting the gyroscope data during emitting the acoustic signal to potential targets. One-second gyro data is 600B. The other data packets only contain gaze direction, network ID and acknowledge respectively. As presented in Section 6.1.3, the delay caused by computation is only about 230ms. In real applications, the wait time is mainly caused by human factors, including the time for paying attention and head nodding. For example, for our social application it is reasonable to require mutual 1s long attention before starting friending. In our smart device case, the attention detection time window is set to 600ms. The time window for attention acquisition can be adjusted for different applications and users. For the social case, before playing acoustic signal, because there is a mutual gaze matching phase, the initiator will be informed to nod his/her head by a beep if there is any coarsely matched users, *i.e.*, someone is looking back at him/her. The reaction time varies between individuals, and in our experiments it is less than 1s for 90% cases. For the smart device case, the initiator just nods head after he/she pays attention to a device. A 1s acoustic signal is sufficient for one head nod. Considering the human factors, in most cases a connection can be built within about 3 seconds from the initiator starts to pay attention to a target.

The connection correctness is determined by the accuracy of the gaze direction and device directions inherently. But the consistency threshold θ determines the false rate of the system output. Fig. 15 shows the false positive and false negative of two cases by changing the setting of the consistency threshold. Here the false positive is the case that a connection is established with a wrong target; and false negative is the case that a small threshold makes the correct target fail to match and no connection is built. We see that, in our two cases, with a proper threshold, false positive is less than 3% while the false negative is about 15%.

7. DISCUSSION AND OPEN ISSUES

The design and instrumentation of a computational smart eyeglasses platform, such as **iGaze**, is extremely complex due to the challenges incurred by noisy sensor data, the mounting and wearing positions of the frame, and uncontrollable user movement. Vi-

sual attention driven based networking, interaction, and control is a barely explored area. While our evaluations demonstrate that **iGaze** is promising, there are a few steps to take before such kind of platforms is ready to practical use with scalability.

Intention Capture and Understanding: In this work, a key component is to capture and understand the user visual intention. This is clearly a daunting task even for human being. Our initial implementation takes advantage of some simplified assumptions such as that user will not move her/his head much when he/she is interested in networking or an object. However, in practice, the head might move around during conversation while the eye might continue to track the interested object. Sometimes, in the gaze direction there may exist multiple objects (*e.g.*, side by side, or one being in front of another). In contrary, multiple users may be interested in a same object, thus, some sort of collision avoidance protocol need to be designed in **VAN**. Furthermore, a user may be interested in a moving object, *e.g.*, moving vehicles and running people. Our platform needs to develop new methods that can successfully address these dynamics. Possible solutions are to incorporate a distance estimation module to estimate the distance between two devices, *e.g.*, the method in [32], incorporate a light-weighted efficient gaze tracking using eye-camera only.

Privacy and Security Considerations: A key factor that may impact the adoption of smart glasses is their privacy and security implications. In this work, our main focus is on the functionality and feasibility of our prototype, so we apply some simple strategies to protect various aspects of privacies. For example, when the gazed target is an object, users may turn off the unidirectional mode to avoid unwanted contact (such as SPAM). However, advanced privacy-preserving protocols can be integrated into our system to achieve the following security-related properties. (1)our system can provide a "do-not-track" setting for users who do not want to be connected by others; (2) various profile matching protocols (*e.g.*, [31]) can be introduced to verify the identity of the users to prevent Man-In-The-Middle attack or impersonate attack; (3) some proof/verification protocols can be leveraged to allow users to verify the correctness of the information provided by other users so that no malicious users can fake their directions. Other device-based key extraction protocols can also be adopted, *e.g.*, [30]. Clearly, security implications that will emerge when smart glasses become popular are not limited to above three, and many of them are very challenging. We leave them as our future works to pursue due to the space limit.

Others: While our current proof-of-concept platform operates at low power with COTS devices, considerable performance improvements are possible by using a hardware-only instrumentation. We note that an equally important design issue is the aesthetic perspective of the **iGaze** design, while aesthetic design aspects are beyond the scope of this work. In future platforms, we have to consider where to place these sensors, speakers, and cameras. We need a graceful tradeoff between functionality and the aesthetic design.

8. RELATED WORK

iGaze is related to existing works in the following areas:

Vision-based augment reality systems. Vision related techniques have been widely applied in augment reality systems. Abe et al. [2] propose a methodology to retrieve real-time information from the web. Ha et al. [8] present a scalable vision-based object tracking method for mobile augmented reality systems. Malik et al. [15] study the problem of vision-based pattern tracking with real-time video feeds. [21] uses a head mounted world camera to model the gaze cone and focuses on finding people looking at the same target using image processing techniques. Without eye tracking, it cannot understand the connection intention of people. It works poorly when people are looking at each other and cannot establish a connection to the visual target either.

Gaze tracking. Gaze-tracking technique is an important part of those vision-based systems. Most of those works use multiple cameras. For example, Newman et al. [18] propose to use two cameras and a face model of the user. Ohno and Mukawa [20] use one camera to track the pose of the head and another infra-red camera to capture the pupil location. In those work, a personalized calibration is needed by seeing some markers on the screen. Shih et al. [26] propose a gaze tracking technique without calibration. At least two cameras and at least two point light sources are needed for this solution. Methods requiring multiple cameras incur high hardware, energy and computation cost for wearable glasses. There are a few single camera based gaze tracking techniques. Nishino and Narya [19] use ellipse models for gaze estimation. Wang et al. [29] determine the gaze direction with an iris geometrical model. Starburst [12] only needs a single infra-red camera, it uses RANSAC fit and the model-based optimization to make higher accurate. Hennessey et al. [9] propose a single-camera eye-tracking system with free head motion. Personalized calibration is also required in all the above methods. For the ease of use, we need to minimize manual calibration, and the computation cost must be further reduced for resource limited wearable glasses.

Devices pairing. In our system, we gather the gaze direction information in order to make it as an evidence for pairing two devices together. Although so many devices pairing methods have been proposed, but our solution is tailored for smart glasses and convenient with the well-collected vision information. In general, pairing methods can be classified into three groups: key sharing, measuring the same local environment data and out-of-band(OOB) authentication. OOB is currently the most promising and practical way. For example, Balfanz et al. [3] use "Pre-Authentication" on location-limited channel to do identity authentication. McCune et al. [17] leverage camera-screen as the OOB channel and the QR code as the evidence to make pairing. There are methods based on audio [6] or accelerometers [16] as the OOB channel. Those approaches require a face-to-face communication between people before initiating a pairing procedure. Some "throw and share" applications have been raised recently. Point&Connect [22] offers a device pairing solution when a user points his/her phone towards the intended target. It understands the target selection by measuring maximum distance change based on acoustic signals. Due to that it is range based and has limited raging accuracy (about 7mm), it requires a larger than 20cm displacement towards the target to achieve 90% correctness. Spartacus [28] uses Doppler effect to start an interaction with a neighboring device through a gesture. It uses FFT to calculate speed which suffers time-frequency resolution problem. Hence, it requires a high speed of phone movement ($2 \sim 6$ m/s). Both the large displacement and speed can hardly be achieved by head-mounted devices. Swadloon [10] also uses Doppler effect to find accurate device direction, but it requires phone movement in a horizontal plane. **iGaze** uses Phase Locked Loop (PLL) to track the relative displacement of two head-mounted speakers at high resolution (far less than 1mm) and it outperforms related work for arbitrary mild head movement.

9. CONCLUSION

In this work, we present **iGaze**, a first-of-its-kind low power smart glass that is designed to support vision driven networking by tracking the gaze in realtime and correctly pairing users' devices when mutual gazes happen. This paper presents a soup-to-nuts instrumentation of such a system, including a complete hardware

prototype, a full implementation of a suite of software for gaze detection, device pairing and networking. We extensively evaluate our system on several subjects. Our results show that **iGaze** supports accurate real time networking by a gaze. Our design is highly significant as it paves a way for more exciting ubiquitous vision driven applications, including interest-targeted mobile advertising, reality-augmented interactive gaming, and behavior-based cyber-physical systems.

10. ACKNOWLEDGMENTS

This research is partially supported by NSFC under Grant No. 61125020, NSFC CERG-61361166009. The research of Li is partially supported by NSF CNS-1035894, NSF ECCS-1247944, NSF ECCS-1343306, NSFC under Grants No. 61170216, No. 61228202. We thank all the reviewers for their valuable comments and helpful suggestions.

11. REFERENCES

[1] *Telecommunication Breakdown; Concepts of communication Transmitted via Software-Defined Radio*.

[2] ABE, N., OOGAMI, W., SHIMADA, A., NAGAHARA, H., AND TANIGUCHI, R. Clickable real world: Interaction with real-world landmarks using mobile phone camera. In *IEEE TENCON* (2010).

[3] BALFANZ, D., SMETTERS, D. K., STEWART, P., AND WONG, H. C. Talking to strangers: Authentication in ad-hoc wireless networks. In *NDSS* (2002).

[4] CORBETTA, M., AKBUDAK, E., CONTURO, T. E., SNYDER, A. Z., OLLINGER, J. M., DRURY, H. A., LINENWEBER, M. R., PETERSEN, S. E., RAICHLE, M. E., VAN ESSEN, D. C., ET AL. A common network of functional areas for attention and eye movements. *Neuron 21*, 4 (1998), 761–773.

[5] ERKELENS, C. J., AND VOGELS, I. M. The initial direction and landing position of saccades. *Studies in Visual Information Processing 6* (1995), 133–144.

[6] GOODRICH, M. T., SIRIVIANOS, M., SOLIS, J., TSUDIK, G., AND UZUN, E. Loud and clear: Human-verifiable authentication based on audio. In *ICDCS* (2006), IEEE.

[7] GOSS, D. A., AND WEST, R. W. *Introduction to the Optics of the Eye*. Butterworth-Heinemann Medical, 2001.

[8] HA, J., CHO, K., AND YANG, H. S. Scalable recognition and tracking for mobile augmented reality. In *SIGGRAPH* (2010), ACM, pp. 155–160.

[9] HENNESSEY, C., NOUREDDIN, B., AND LAWRENCE, P. A single camera eye-gaze tracking system with free head motion. In *ETRA* (2006), ACM.

[10] HUANG, W., XIONG, Y., LI, X., LIN, H., MAO, X., YANG, P., AND LIU, Y. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. In *Infocom* (2014), IEEE.

[11] KOHLBECHER, S., BARDINST, S., BARTL, K., SCHNEIDER, E., POITSCHKE, T., AND ABLASSMEIER, M. Calibration-free eye tracking by reconstruction of the pupil ellipse in 3d space. In *ETRA* (2008), ACM.

[12] LI, D., WINFIELD, D., AND PARKHURST, D. J. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *CVPR Workshops* (2005), IEEE, pp. 79–79.

[13] LI, Z., WANG, C., JIANG, C., AND LI, X.-Y. Lass: Local-activity and social-similarity based data forwarding in mobile social networks. *TPDS* (2013).

[14] LIVERSEDGE, S. P., AND FINDLAY, J. M. Saccadic eye movements and cognition. *Trends in cognitive sciences 4*, 1 (2000), 6–14.

[15] MALIK, S., MCDONALD, C., AND ROTH, G. Hand tracking for interactive pattern-based augmented reality.

[16] MAYRHOFER, R., AND GELLERSEN, H. Shake well before use: Authentication based on accelerometer data. In *Pervasive computing*. Springer, 2007, pp. 144–161.

[17] MCCUNE, J. M., PERRIG, A., AND REITER, M. K. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *S&Privacy* (2005), IEEE.

[18] NEWMAN, R., MATSUMOTO, Y., ROUGEAUX, S., AND ZELINSKY, A. Real-time stereo tracking for head pose and gaze estimation. In *International Conference on Automatic Face and Gesture Recognition* (2000), IEEE, pp. 122–128.

[19] NISHINO, K., AND NAYAR, S. K. Eyes for relighting. In *ACM Transactions on Graphics* (2004), vol. 23, pp. 704–711.

[20] OHNO, T., AND MUKAWA, N. A free-head, simple calibration, gaze tracking system that enables gaze-based interaction. In *ETRA* (2004), ACM.

[21] PARK, H. S., JAIN, E., AND SHEIKH, Y. 3d social saliency from head-mounted cameras. In *Advances in Neural Information Processing Systems* (2012), pp. 431–439.

[22] PENG, C., SHEN, G., ZHANG, Y., AND LU, S. Point&connect: intention-based device pairing for mobile phone users. In *MobiSys* (2009), ACM, pp. 137–150.

[23] RICE, M. *Digital Communications: A Discrete-Time Approach*. Prentice Hall, 2008.

[24] SALVUCCI, D. D., AND GOLDBERG, J. H. Identifying fixations and saccades in eye-tracking protocols. In *ETRA* (2000).

[25] SEN, T., AND MEGAW, T. The effects of task variables and prolonged performance on saccadic eye movement parameters. *Advances in Psychology 22* (1984), 103–111.

[26] SHIH, S.-W., WU, Y.-T., AND LIU, J. A calibration-free gaze tracking technique. In *International Conference on Pattern Recognition* (2000), vol. 4, IEEE, pp. 201–204.

[27] STERN, H. The significance of impulse buying today. *The Journal of Marketing* (1962), 59–62.

[28] SUN, Z., PUROHIT, A., BOSE, R., AND ZHANG, P. Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing. In *MobiSys* (2013).

[29] WANG, J., SUNG, E., AND VENKATESWARLU, R. Eye gaze estimation from a single image of one eye. In *International Conference on Computer Vision* (2003), IEEE, pp. 136–143.

[30] XI, W., LI, X.-Y., QIAN, C., HAN, J., TANG, S., ZHAO, J., AND ZHAO, K. Keep: Fast secret key extraction protocol for d2d communication. In *IWQoS* (2014), IEEE.

[31] ZHANG, L., LI, X.-Y., AND LIU, Y. Message in a sealed bottle: Privacy preserving friending in social networks. In *ICDCS* (2013), IEEE.

[32] ZHANG, L., LIU, K., JIANG, Y., LI, X.-Y., LIU, Y., AND YANG, P. Montage: Combine frames with movement continuity for realtime multi-user tracking. In *Infocom* (2014), IEEE.